Date=02/07/2020
Lecture By=Shubham Joshi
Notes By=Upadhyay Hemanshu

IN PREVIOUS LECTURE**(QUICK RECAP)**	**IN TODAY'S LECTURE(OVERVIEW)**
=LEARNED ABOUT **GIT COMMANDS**	Introduction to Github,Master Branch&Dev Branch and Fork
=1.git init 2.git status 3.git add **=4.git commit 5.git log 6.git diff**	**New Commands**=Gitpush,Gitpull,Gitmerge, Gitclone,Gitbrandev,Gitcheckot -b, Gitcheckout hash of commit

=In this class we are gonna look into **Github** and learn How to **clone** your Repository And **Push** to Github

== **Introduction** To the **Github**

-**GitHub** is a Git repository hosting service

-**Github** is owned by **Microsoft**

-Github saves your data to their **Servers/Cloud** and when you or anyone needs your data, it **clones** your data/file to your Computer. =Site Address-> Github

==About **Master Branch**

-**Sole branch** of repository is Known As Master Branch

-Master branch keeps your all code releases.

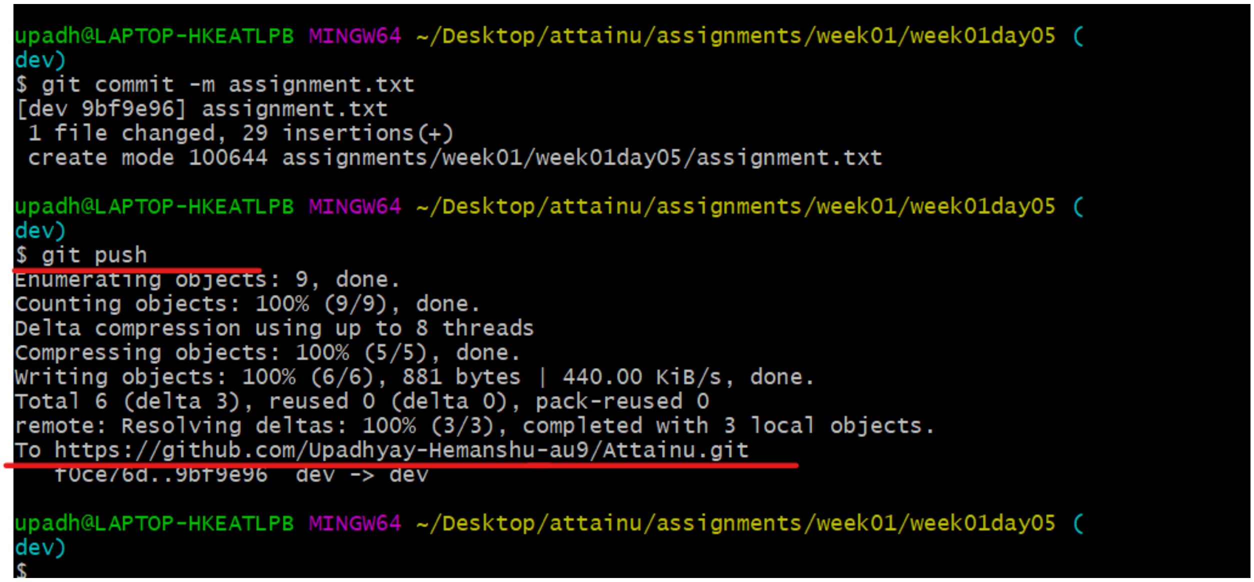
==We also learned some **new commands** in this lectures that are explained below

1.Git push

-The **git push command** is used to upload local repository content to a Cloud/Servers repository.

-**Pushing** is how you transfer commits from your local repository to a remote repo.

- To Push file you have to to **type git push**(Git push only works when you **Commit** anyfile without **committing** this command won't work)



```
upadh@LAPTOP-HKEATLPB MINGW64 ~/Desktop/attainu/assignments/week01/week01day05 (
dev)
$ git commit -m assignment.txt
[dev 9bf9e96] assignment.txt
 1 file changed, 29 insertions(+)
 create mode 100644 assignments/week01/week01day05/assignment.txt

upadh@LAPTOP-HKEATLPB MINGW64 ~/Desktop/attainu/assignments/week01/week01day05 (
dev)
$ git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 881 bytes | 440.00 KiB/s, done.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/Upadhyay-Hemanshu-au9/Attainu.git
   f0ce76d..9bf9e96  dev -> dev

upadh@LAPTOP-HKEATLPB MINGW64 ~/Desktop/attainu/assignments/week01/week01day05 (
dev)
$
```

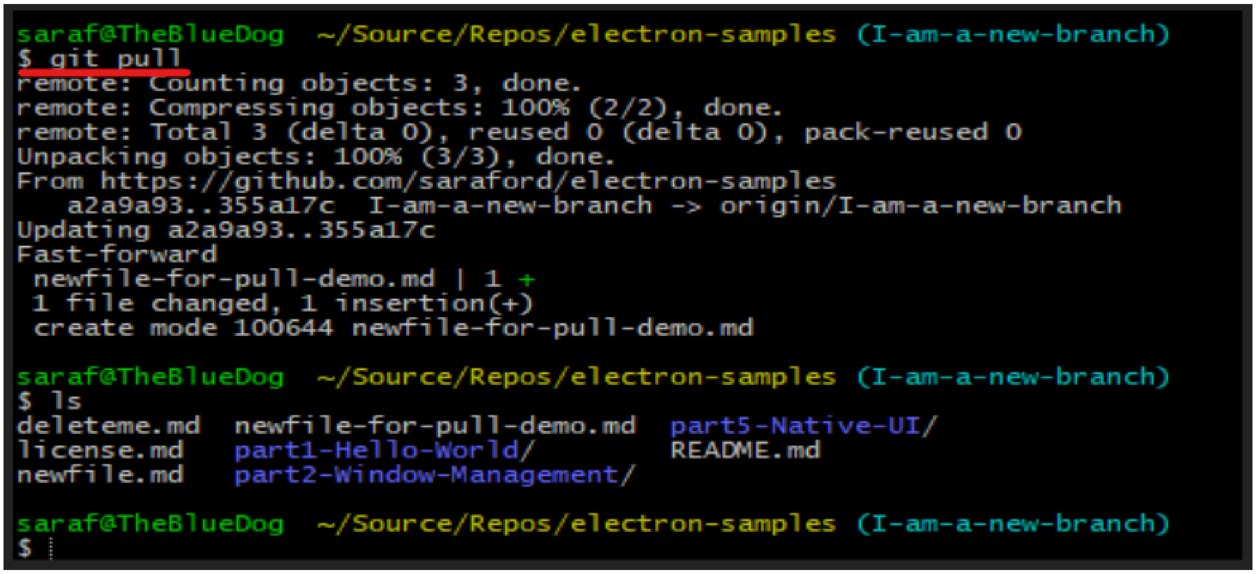
TO Know More About Gitpush "[Click Here](#)"

2.Git pull

-The **git pull command** is used to fetch and download content from a remote repository

-it is also used to Merge Changes to your local repository.

-The git pull command is actually a combination of two other commands, **git fetch** followed by **git merge**.

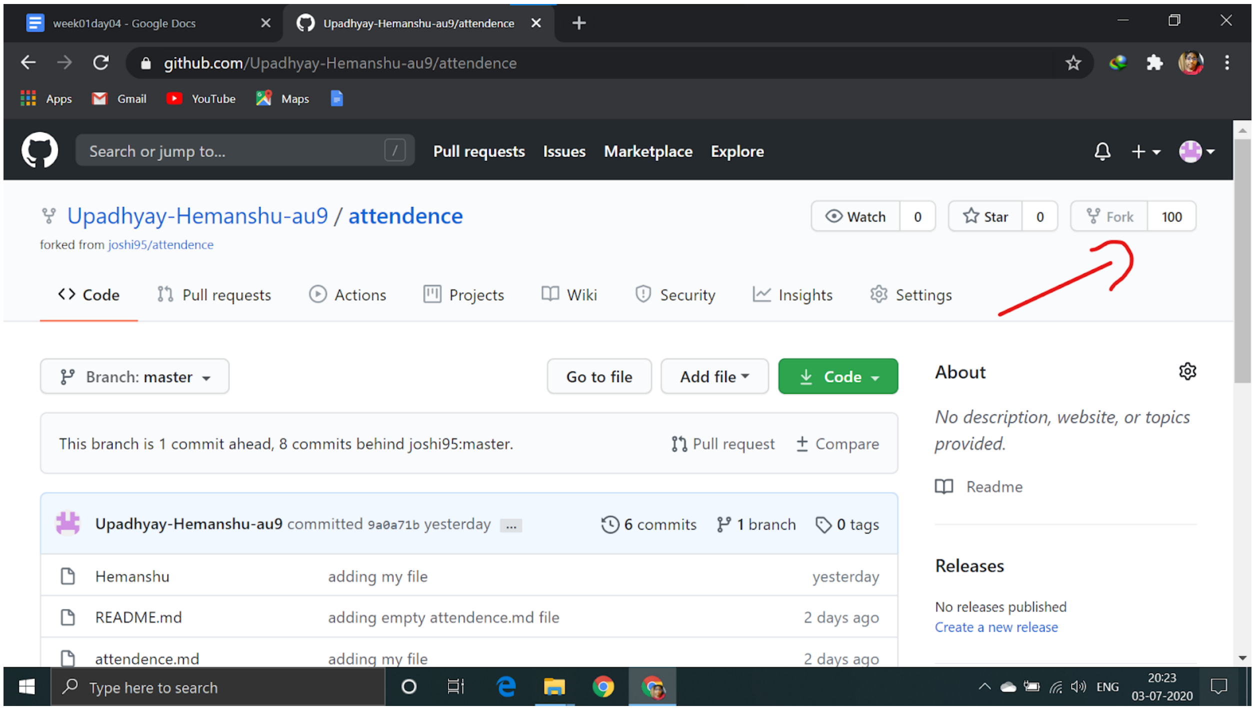


```
saraf@TheBlueDog  ~/Source/Repos/electron-samples (I-am-a-new-branch)
$ git pull
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/saraford/electron-samples
   a2a9a93..355a17c  I-am-a-new-branch -> origin/I-am-a-new-branch
Updating a2a9a93..355a17c
Fast-forward
 newfile-for-pull-demo.md | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 newfile-for-pull-demo.md

saraf@TheBlueDog  ~/Source/Repos/electron-samples (I-am-a-new-branch)
$ ls
deleteme.md  newfile-for-pull-demo.md  part5-Native-UI/
license.md   part1-Hello-World/        README.md
newfile.md   part2-Window-Management/

saraf@TheBlueDog  ~/Source/Repos/electron-samples (I-am-a-new-branch)
$ 
```

To Know More About Gitpull "Click Here"

3.Fork

-**Git Fork** means you just create a copy of the main repository of a project source code to your own **GitHub** profile.
-in Fork you can experiment whatever you like without affecting the main source of that project. .



Upadhyay-Hemanshu-au9 / attendence

forked from joshi95/attendence

This branch is 1 commit ahead, 8 commits behind joshi95:master.

Upadhyay-Hemanshu-au9 committed 9a0a71b yesterday

6 commits 1 branch 0 tags

Hemanshu	adding my file	yesterday
README.md	adding empty attendence.md file	2 days ago
attendence.md	adding my file	2 days ago

About

No description, website, or topics provided.

Readme

Releases

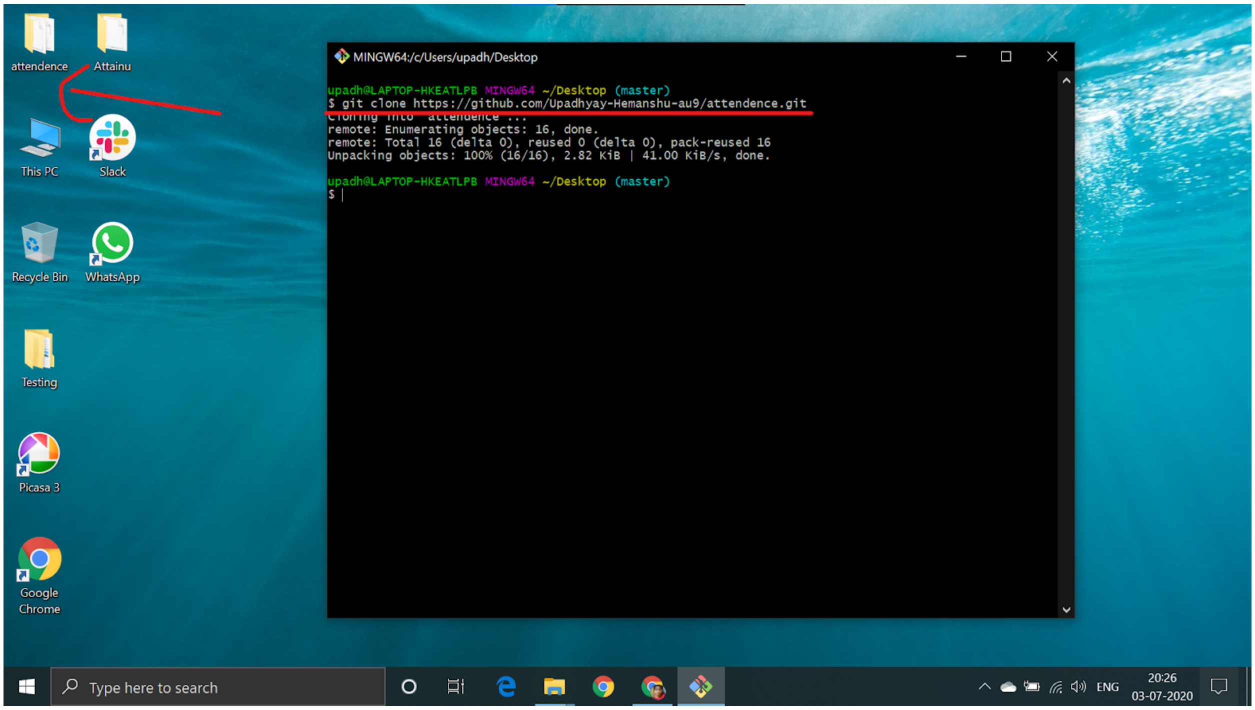
No releases published

Create a new release

4.Git Clone

-**git clone** is a **Git** command line utility which is used to target an existing repository and create a **clone**, or copy of the target repository.

-To create gitclone you just have to type “git clone <url of git repository>”



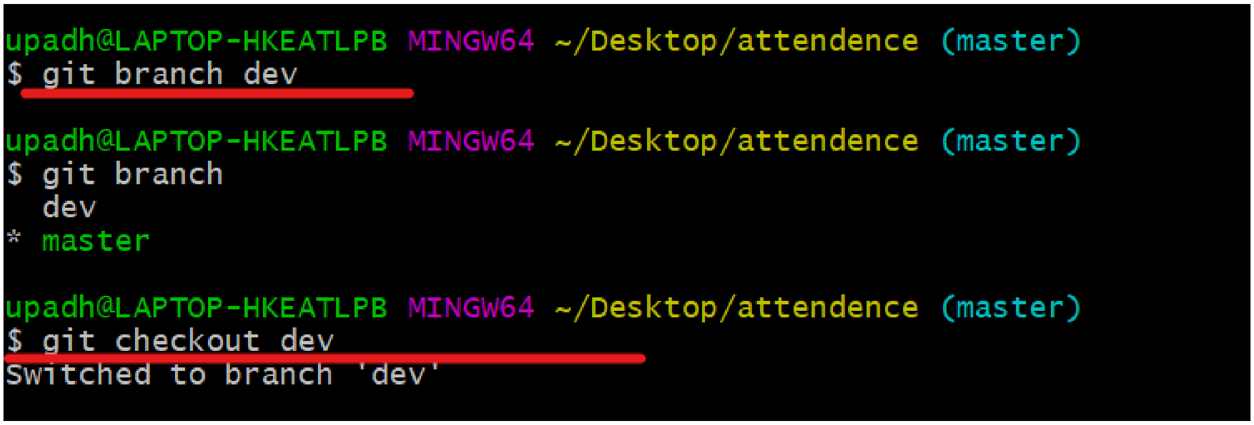
```
upadh@LAPTOP-HKEATLPB MINGW64 ~/Desktop (master)
$ git clone https://github.com/Upadhyay-Hemanshu-au9/attendence.git
Cloning into 'attendence'...
remote: Enumerating objects: 16, done.
remote: Total 16 (delta 0), reused 0 (delta 0), pack-reused 16
Unpacking objects: 100% (16/16), 2.82 KiB | 41.00 KiB/s, done.

upadh@LAPTOP-HKEATLPB MINGW64 ~/Desktop (master)
$ |
```

To know More About GitClone “Click Here”

5.git branch dev

-It is used to create new Branch in your Gitbash
To create dev(developer) just type “git branch dev” and done.
[Note-this command will only work if you are in some repository if you type this in the beginning/desktop this will not work**]**

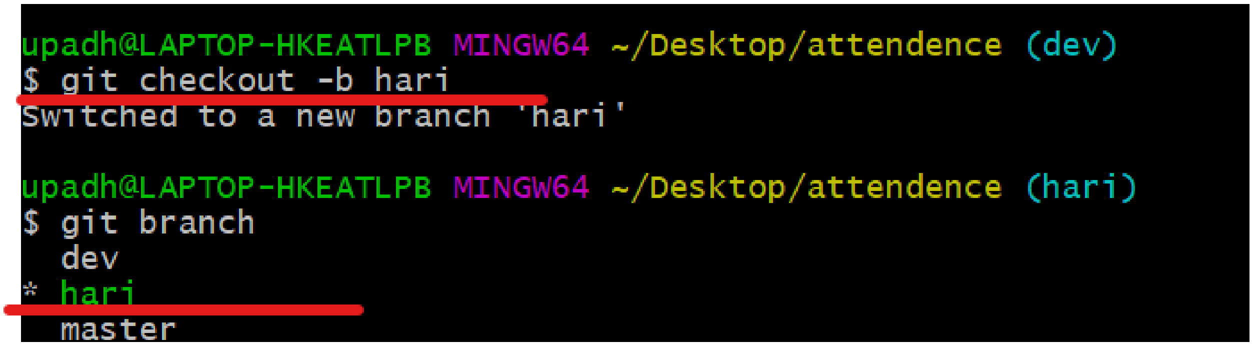
==To know which branch in you are or want to know how many branch are there just type git branch,to change branch type git checkout dev



```
upadh@LAPTOP-HKEATLPB MINGW64 ~/Desktop/attendence (master)
$ git branch dev

upadh@LAPTOP-HKEATLPB MINGW64 ~/Desktop/attendence (master)
$ git branch
  dev
* master

upadh@LAPTOP-HKEATLPB MINGW64 ~/Desktop/attendence (master)
$ git checkout dev
Switched to branch 'dev'
```

6.git checkout -b

- it is used to create and change directory at same time
-It means this command can create a new Branch and will change the Branch At the same time



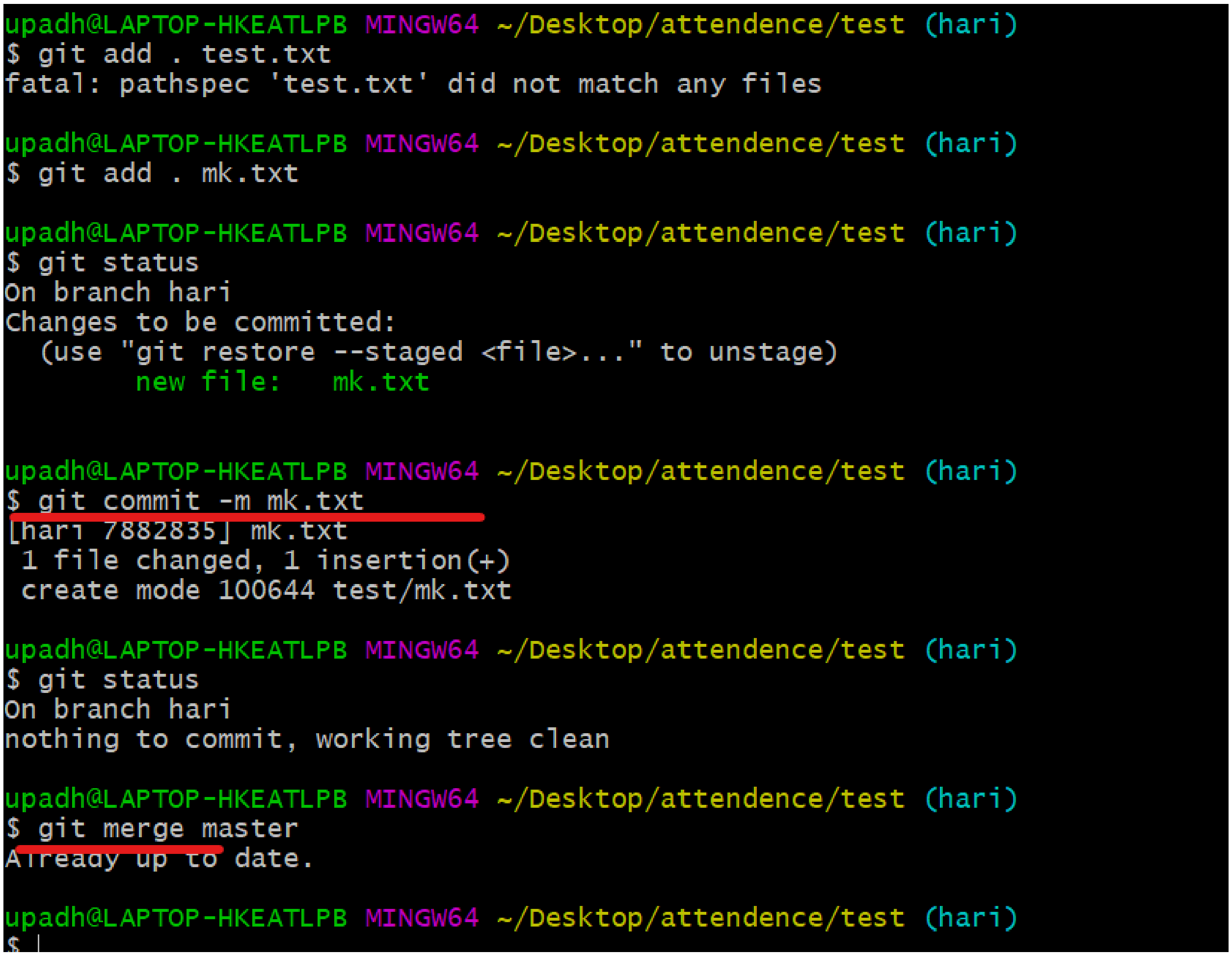
```
upadh@LAPTOP-HKEATLPB MINGW64 ~/Desktop/attendence (dev)
$ git checkout -b hari
Switched to a new branch 'hari'

upadh@LAPTOP-HKEATLPB MINGW64 ~/Desktop/attendence (hari)
$ git branch
  dev
* hari
  master
```

To Know More About Git Branchs “Click Here”

7.Git merge

-git merge basically merges your dev branch’s file into master branch

-to merge files type Git merge <branch name> {in which you would like to merge}



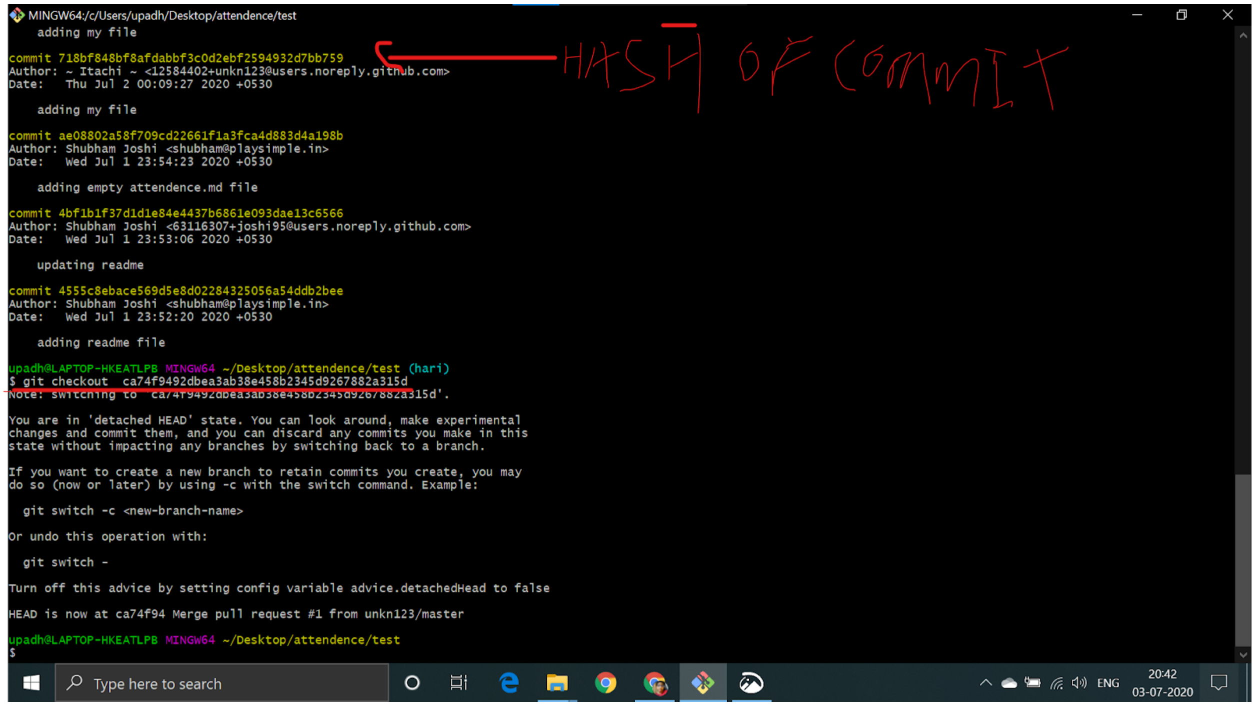
```
upadh@LAPTOP-HKEATLPB MINGW64 ~/Desktop/attendence/test (hari)
$ git add . test.txt
fatal: pathspec 'test.txt' did not match any files

upadh@LAPTOP-HKEATLPB MINGW64 ~/Desktop/attendence/test (hari)
$ git add . mk.txt

upadh@LAPTOP-HKEATLPB MINGW64 ~/Desktop/attendence/test (hari)
$ git status
On branch hari
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   mk.txt


upadh@LAPTOP-HKEATLPB MINGW64 ~/Desktop/attendence/test (hari)
$ git commit -m mk.txt
[hari 7882835] mk.txt
 1 file changed, 1 insertion(+)
 create mode 100644 test/mk.txt

upadh@LAPTOP-HKEATLPB MINGW64 ~/Desktop/attendence/test (hari)
$ git status
On branch hari
nothing to commit, working tree clean

upadh@LAPTOP-HKEATLPB MINGW64 ~/Desktop/attendence/test (hari)
$ git merge master
Already up to date.

upadh@LAPTOP-HKEATLPB MINGW64 ~/Desktop/attendence/test (hari)
$
```

To Know More About Git Merge “Click Here”

8.git checkout hash of commit

-this command helps you to fix your issues/mistakes that you have done in previous commit’s

-Just type Git checkout hash of commit which is shown below



```
    adding my file

commit 718bf848bf8afdabbf3c0d2ebf2594932d7bb759
Author: ~ Itachi ~ <12584402+unkn123@users.noreply.github.com>
Date:   Thu Jul 2 00:09:27 2020 +0530

    adding my file

commit ae08802a58f709cd22661f1a3fca4d883d4a198b
Author: Shubham Joshi <shubham@playsimple.in>
Date:   Wed Jul 1 23:54:23 2020 +0530

    adding empty attendence.md file

commit 4bf1b1f37d1d1e84e4437b6861e093dae13c6566
Author: Shubham Joshi <63116307+joshi95@users.noreply.github.com>
Date:   Wed Jul 1 23:53:06 2020 +0530

    updating readme

commit 4555c8ebace569d5e8d02284325056a54ddb2bee
Author: Shubham Joshi <shubham@playsimple.in>
Date:   Wed Jul 1 23:52:20 2020 +0530

    adding readme file

upadh@LAPTOP-HKEATLPB MINGW64 ~/Desktop/attendence/test (hari)
$ git checkout  ca74f9492dbea3ab38e458b2345d9267882a315d
Note: switching to 'ca74f9492dbea3ab38e458b2345d9267882a315d'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at ca74f94 Merge pull request #1 from unkn123/master

upadh@LAPTOP-HKEATLPB MINGW64 ~/Desktop/attendence/test
$
```

To Know More About It “Click Here”